REMARKS/ARGUMENTS

The Examiner in an advisory action dated 5/12/2004 indicated that proposed amendments submitted by the applicant on 4/28/2004 would not be entered because they raised new issues that required further consideration and/or search.

On the basis of an interview conducted on 2 PM (EST), April 22, 2004, in which the applicants discussed the claims with the Examiner and the Examiner's supervisor, the applicants had rewritten dependent claims 4, 17, 23, 36, 42, 55 in independent form in the amendment that was not entered, such that, the rewritten dependent claims included all the limitations of the base and intervening claims on which they depended. Applicants understood from the Examiner that the rewritten dependent claims 4, 17, 23, 36, 42, 55 were likely to patentable over the cited Leymann (US 6,065,009). According to the Applicant, the rewritten dependent claims 4, 17, 23, 36, 42, 55 did not raise new issues and should have been entered by the Examiner. Nevertheless, the applicant is submitting the current amendment with a Request for Continued Examination.

In the current paper the applicants have added eighteen new claims numbered 58-75. Claims 1-57 are the same as in the amendment of 4/28/2004 that was not entered by the Examiner. The Examiner is again requested to note that the Examiner and the Examiner's supervisor had indicated during the interview on April 22, 2004 that claims 4, 17, 23, 36, 42, 55 were likely to patentable over the cited Leymann (US 6,065,009).

The Examiner rejected claims 1-57 under 35 U.S.C. 102 as being anticipated by Leymann (US 6,065,009) [Note: Examiner has mistyped the patent number of Leymann as US 6,065,099].

Applicants traverse the claim rejections and maintain the patentability of claims 1-57.

Claims 1, 20, 39

Claims 1, 20, 39 provide a method, system and article of manufacture for maintaining workflow related information, comprising:

providing at least one table in a database storing workflow related data;

providing a plurality of programming interfaces, wherein each programming interface specifies an operation to perform on the workflow related data in the at least one table, wherein each programming interface is associated with one stored procedure call;

providing in the database one stored procedure for each stored procedure call and corresponding method, wherein the one stored procedure includes a plurality of database statements to perform the programming interface operation; and

executing the one stored procedure in the database to perform the corresponding programming interface operation on workflow related data in one table.

The Examiner has rejected claim 1 under 35 U.S.C. 102 as being anticipated by Leymann Applicants traverse.

Nowhere does the cited Lehman (col. 6: lines 14-21, col. 15: lines 19-25, col. 11: lines 21-23; col. 11: lines 21-28) teach or disclose the claim requirement of providing in the database one stored procedure for each stored procedure call and corresponding method in combination with the other claim limitations.

The Examiner has cited col. 11 of lines 21-28 of Leymann as disclosing the claims requirement of providing in the database one stored procedure for each stored procedure call and corresponding method. Col. 11, lines 21-28 of the cited Leymann is as follows: "a process instance is started either via the graphical user interface, or via a callable process application programming interface. When a process is started, the start activities are located, the proper people are determine, and the activities are posted onto the work list of the selected people. If a use selects the activity, the activity is executed and removed from the work list of any other user to whom the activity has been posted." Therefore, the cited Leymann discusses how an instance of a process is started via an programming interface. The cited Leymann further discusses that when a process is started, the corresponding start activities are posted in a work list. Methods to remove activities from the work list are discussed by the cited Leymann. The cited Leymann discusses starting a process but nowhere does the cited Leymann teach or disclose the claim

requirement of providing in the database one stored procedure for each stored procedure call and corresponding method in combination with the other claim limitations.

Additional reasons for the patentablity of claims 1, 20, 39 are given below:

The claims require providing in the database one stored procedure, wherein the one stored procedure includes a plurality of database statements to perform the programming interface operation, and executing the one stored procedure in the database to perform the corresponding programming interface operation on workflow related data in one table.

The cited Leymann (col. 6: lines 14-21; col. 11: lines 21-28; col. 15: lines 19-25) discusses implementations of events as activities in process models of workflow management systems. The Examiner has cited Leymann (col. 11, lines 21-28) as disclosing the claim requirement of providing in the database one stored procedure for each stored procedure call and corresponding method, wherein the one stored procedure includes a plurality of database statements to perform the programming interface operation, and executing the one stored procedure in the database to perform the corresponding programming interface operation on workflow related data in one table. Applicants traverse.

The cited Leymann (col. 11, lines 21-28) discusses a process instance as being started by a callable programming interface. The cited Leymann (col. 11, lines 21-28) further discusses the steps that occur when the process instance is started. Therefore, the discussion in the cited Leymann (col. 11, lines 21-28) is related to the starting of a process instance by a callable programming interface and the steps that occur when the process instance is started. Nowhere does the cited Leymann (col. 11, lines 21-28) teach or disclose the claim requirement of providing in the database one stored procedure, wherein the one stored procedure includes a plurality of database statements to perform the programming interface operation, and executing the one stored procedure in the database to perform the corresponding programming interface operation on workflow related data in one table.

For the above reasons, claims 1, 20 and 39 are patentable over the cited art.

Claims 13, 32, 51

Claims 13, 32, and 51 provide a method, system, and article of manufacture for maintaining information in a database, comprising:

(a) receiving column definitions for multiple columns in at least one table; and

(b) for each table for which column definitions are received, performing:

(i) generating a table in the database including one column for each column definition, wherein each column is generated with attributes specified by the column definition for which the column is generated; and

(ii) generating at least one stored procedure including database statements in the database to perform an operation on the data in the generated table, wherein the stored procedure is capable of accessing the columns generated according to the column definitions.

The Examiner has rejected claim 13 under 35 U.S.C. 102 as being anticipated by Leymann. Applicants traverse.

The claims require for each table for which column definitions are received, generating at least one stored procedure including database statements in the database to perform an operation on the data in the generated table, wherein the stored procedure is capable of accessing the columns generated according to the column definitions.

The cited Leymann (col. 14: lines 22-24, 36-39; col. 11: lines 21-28) discusses tracking of events in an event table. The event tables discussed in the cited Leymann (FIG. 4) include a process instance identifier, an event identifier, an event Name and input containers as columns of the table.

The Examiner has cited Leymann (col. 14, lines 22-24, 36-39) as disclosing the claim requirement of generating a table in the database including one column for each column definition, wherein each column is generated with attributes specified by the column definition for which the column is generated. The cited Leymann discusses tracking of events in an event table. The event tables discussed in the cited Leymann (FIG. 4) include a process instance identifier, an event identifier, an event Name and input containers as columns of the table.

Nowhere does the cited Leymann (col. 14, lines 22-24, 36-39) teach or disclose the claim requirement of generating a table in the database including one column for each column definition, wherein each column is generated with attributes specified by the column definition for which the column is generated. Specifying process instances that might consume event instances as mentioned by the Examiner does not teach or disclose the claim requirement of generating a table in the database including one column for each column definition, wherein each column is generated with attributes specified by the column definition for which the column is generated.

Additionally, the Examiner mentions that the cited Leymann (col. 11: lines 21-28) discusses the claim requirement of generating at least one stored procedure including database statements in the database to perform an operation on the data in the generated table, wherein the stored procedure is capable of accessing the columns generated according to the column definitions. The cited Leymann (col. 11, lines 21-28) discusses a process instance as being started by a callable programming interface. The cited Leymann (col. 11, lines 21-28) further discusses the steps that occur when the process instance is started. Therefore, the discussion in the cited Leymann (col. 11, lines 21-28) is related to the starting of a process instance by a callable programming interface and the steps that occur when the process instance is started. Nowhere does the cited Leymann (col. 11, lines 21-28) teach or disclose the claim requirement of generating at least one stored procedure including database statements in the database to perform an operation on the data in the generated table, wherein the stored procedure is capable of accessing the columns generated according to the column definitions.

For the above reasons, claims 1, 20 and 39 are patentable over the cited art.

Claims 2-12, 21-31, 40-50

The Examiner has also rejected pending claims 2-12, 21-31, 40-50 that depend on the pending independent amended claims 1, 20, 39 respectively. Applicants submit that these claims are patentable over the cit d art because they depend from claims 1, 11, and 21 respectively

which are patentable over the cited art for the reason discussed above, and because the combination of the limitations in the dependent claims 2-12, 21-31, 40-50 and the base and intervening claims from which they depend provide further grounds of distinction over the cited art. Moreover, the following of these claims provide additional grounds of patentability over the cited art for the reasons discussed below.

Claims 14-19, 33-38, 52-57

The Examiner has also rejected pending claims 14-19, 33-38, 52-57 that depend on the pending independent amended claims 13, 32, 51 respectively. Applicants submit that these claims are patentable over the cited art because they depend from claims 13, 32, 51 respectively which are patentable over the cited art for the reason discussed above, and because the combination of the limitations in the dependent claims 14-29, 33-38, 52-57 and the base and intervening claims from which they depend provide further grounds of distinction over the cited art. Moreover, the following of these claims provide additional grounds of patentability over the cited art for the reasons discussed below.

Currently amended claims 4, 23, and 42

Original claim 4 has been amended to include all the limitations of claim 1 that were not already present in original claim 4. Amended claim 4 is a method for maintaining workflow related information, comprising:

providing a plurality of tables including workflow related data;

providing a set of programming interfaces for each table including workflow related data, wherein each set of programming interfaces defines a same set of operations to perform on the table for which the set is provided, wherein each programming interface is associated with one stored procedure call; and

providing stored procedure calls and stored procedures in the database for each set of programming interfaces to implement programming interface operations in the set, wherein one

stored procedure is provided for each stored procedure call and corresponding method, and wherein the one stored procedure includes a plurality of database statements to perform a programming interface operation; and

executing the one stored procedure in the database to perform the corresponding programming interface operation on workflow related data in one table.

The Examiner has rejected claim 4, based on the cited Leymann (col. 11, lines 21-23) which is as follows: "This allows to make changes to the process model without affecting executing process instances. A process instance is started either via the graphical interface, or via the callable process application interface" Therefore, the cited Leymann (col. 11, lines 21-23) discusses starting a process instance via a callable process application programming interface. Nowhere does the cited Leymann teach or discuss the claim requirement of providing stored procedure calls and stored procedures in the database for each set of programming interfaces to implement the programming interface operations in the set, wherein one stored procedure is provided for each stored procedure call and corresponding method, and wherein the one stored procedure includes a plurality of database statements to perform a programming interface operation. Additionally, since claim 4 includes the limitations of claim 1, the reasons provided for the patentablity of claim 1 are also applicable to claim 4.

Therefore claims 4, 23, and 42 are patentable over the cited art because the cited Leymann do not teach or disclose all the claim limitations.

Claims 5, 24, and 43

Claim 5 depends on dependent claim 4, wherein the programming interfaces for each table are members of a table class, wherein there is one table class for each table including workflow related data.

The Examiner has rejected claim 5 based on the analysis of claim 1. However, claim 5 includes additional limitations beyond those of claim 1. The cited Leymann discusses that the programming interfaces are called to start process instances. Nowhere does the cited Leymann

teach or disclose the claim requirement that the programming interfaces for each table are members of a table class, wherein there is one table class for each table.

Therefore, claims 5, 24, and 43 are patentable over the cited art.

Claims 6, 25, and 44

Claim 6 depends on claim 4, wherein the workflow related tables comprise:

a workflow file table, wherein each entry includes one workflow file including code defining a workflow and nodes of the workflow;

a worklist table, wherein each entry includes a worklist associated with a plurality of work items performed at the nodes defined in the associated workflow files; and

an action list table, wherein each entry includes a list of actions capable of being performed at the nodes defined within one workflow file in the workflow file table.

The cited Leymann (col. 6: lines 14-21, lines 19-26; col. 5: lines 33-42) discusses event tables and event generators. Nowhere does the cited Leymann teach or disclose the claim requirement that the workflow related tables comprise a workflow file table, wherein each entry includes one workflow file including code defining a workflow and nodes of the workflow; a worklist table, wherein each entry includes a worklist associated with a plurality of work items performed at the nodes defined in the associated workflow files; and an action list table, wherein each entry includes a list of actions capable of being performed at the nodes defined within one workflow file in the workflow file table.

Therefore claims 6, 25 and 44 are patentable over the cited art.

Claims 7, 26, and 45

Claim 7 depends on claim 6, wherein the workflow related tables further comprise:

an action table, wherein each entry includes one action name and an address of a program implementing the action, wherein every action listed in the entries of the action list table are defined in th action table.

The cited Leymann (col. 15: lines 19-25) discusses an event monitor application programming interface to allow applications to query, remove entries, insert entries into the event table. Nowhere does the cited Leymann teach or disclose the claim requirement of an action table, wherein each entry includes one action name and an address of a program implementing the action, wherein every action listed in the entries of the action list table are defined in the action table.

Therefore claims 7, 26 and 45 are patentable over the cited art.

Claims 8, 27, 46

Claim 8 depends on claim 4, wherein each set of programming interfaces includes:

a create programming interface to invoke one stored procedure to create an entry of workflow related data in the table associated with the set;

a get programming interface to invoke one stored procedure to retrieve one entry of workflow related data from the table associated with the set;

an update programming interface to invoke one stored procedure to update one entry of workflow related data in the table associated with the set;

a delete programming interface to invoke one stored procedure to delete one entry of workflow related data in the table associated with the set; and

a list programming interface to invoke one stored procedure to list a description of all entries of workflow related data in the table associated with the set.

The cited Leymann (col. 15: lines 19-22, 21-23, 24-25) discusses programming interfaces for performing various functions such as querying, removing, etc. from tables for monitoring events. The cited Leymann does not the claim requirement of programming interfaces that invoke one stored procedure as required by the claims.

Therefore, claims 8, 27, 46 are patentable over the cited art.

Claims 9, 28, 47

Claim 9 depends on claim 1, and further comprises:

receiving workflow related data from a workflow builder program, wherein the workflow builder program calls the programming interfaces to invoke the associated stored procedure calls and the stored procedures to perform the programming interface operations on the at least one table of workflow related data.

The cited Leymann (col. 11: lines 21-28) discusses an event monitor programming interface for querying, removing entries, inserting entries into a table. The cited Leymann also discusses that an event has properties that include the name of the program implementing the event generator and the name of the event and an indicator. However, nowhere does the cited Leymann teach or disclose the claim requirement of the workflow builder program calling the programming interfaces.

Therefore claims 9, 28, 47 are patentable over the cited art.

Claims 11, 30, 49

Claim 11 depends on claim 10, and further comprises:

executing one stored procedure to retrieve one workflow file in response to receiving a stored procedure call invoked by a programming interface from the buildtime program to retrieve one workflow file, wherein the build time program exports the retrieved workflow file to a workflow engine.

The cited Leymann (cols 15, line 51 - col. 16: line 10) discusses additions to the flowmark definition language and data structures for event identification and response information. Nowhere does the cited Leymann teach or disclose the claim requirement of executing one stored procedure to retrieve one workflow file in response to receiving a stored procedure call invoked by a programming interface from the buildtime program to retrieve one workflow file, wherein the build time program exports the retrieved workflow file to a workflow engine.

Therefore claims 11, 30, 49 are patentable over the cited art.

Claims 15, 34, 53

Claim 15 depends on claim 14, wherein the provided programming interfaces are members of an object oriented class, wherein one class is provided for each generated table, and wherein the programming interfaces for one class are capable of invoking stored procedures to perform database operations on the table for which the class of programming interfaces are provided.

The cited Leymann (col. 11: lines 21-23) discusses starting a process instance by a callable application program interface or via a graphical programming interface. Nowhere does the cited Leymann teach or disclose the claim requirement that the provided programming interfaces are members of an object oriented class, wherein one class is provided for each generated table, and wherein the programming interfaces for one class are capable of invoking stored procedures to perform database operations on the table for which the class of programming interfaces are provided.

Therefore claims 15, 34, and 53 are patentable over the cited art.

Claims 16, 35, and 54

Claim 16 depends on claim 13 and further comprises providing one programming interface for each generated stored procedure, wherein each provided programming interface is called to invoke one stored procedure call to further invoke and execute one stored procedure in the database.

The cited Leymann (col. 11: lines 21-28) discusses that a process instance is started by a callable process application programming interface. The cited Leymann further discusses what happens after a process instance starts in terms of workflow. Nowhere does the cited Leymann teach or disclose the claim requirement for providing one programming interface for each

generated stored procedure, wherein each provided programming interface is called to invoke one stored procedure call to further invoke and execute one stored procedure in the database.

Therefore, claims 16, 35 and 54 are patentable over the cited art.

Currently amended claims 17, 36, 55

Claim 17 has been rewritten in independent form, to include all the limitations of original claim 17 and the limitations of claims 13 and 16 on which claim 17 depended. Currently amended claim 17 is a method for maintaining information in a database, comprising:

(a) receiving column definitions for multiple columns in at least one table; and

(b) for each table for which column definitions are received, performing:

(i) generating a table in the database including one column for each column definition, wherein each column is generated with attributes specified by the column definition for which the column is generated; and

(ii) generating at least one stored procedure including database statements in the database to perform an operation on the data in the generated table, wherein the stored procedure is capable of accessing the columns generated according to the column definitions;

(c) providing one programming interface for each generated stored procedure, wherein each provided programming interface is called to invoke one stored procedure call to further invoke and execute one stored procedure in the database, wherein a set of programming interfaces is provided for each table, wherein each set of programming interfaces includes:

(i) a create programming interface to invoke one stored procedure to create an entry in the table associated with the set;

(ii) a get programming interface to invoke one stored procedure to retrieve data from the table associated with the set;

(iii) an update programming interface to invoke one stored procedure to update one entry in the table associated with the set;

(iv) a delete programming interface to invoke one stored procedure to delete one entry in the table associated with the set; and

(v) a list programming interface to invoke one stored procedure to list a description of all entries in the table associated with the set.

The Examiner has rejected claim 17 based on the cited Leymann (col. 15: lines 19-22, 21-23, 24-25, where lines 19-25 of Leymann that includes the cited Leymann is as follows: "Event Monitor Programming Interface: The event monitor provides an application programming interface to allow applications to request event monitor functions. The set of functions, including requests, such as querying the posted event table, removing entries from the posted event table, querying the awaited event table, removing entries form the awaited event table, and inserting entries into the posted event table." Therefore, The cited Leymann (col. 15: lines 19-22, 21-23, 24-25) discusses programming interfaces for performing various functions such as querying, removing, etc. from tables for monitoring events. The cited Leymann does not the claim requirement of providing one programming interface for each generated stored procedure, wherein each provided programming interface is called to invoke one stored procedure call to further invoke and execute one stored procedure in the database.

Therefore, claims 17, 36, 55 are patentable over the cited art.

Conclusion

For all the above reasons, Applicant submits that the pending claims 1-75 are patentable over the art of record. Applicants have indicated appropriate fees for claims. Nonetheless, should any additional fees be required, please charge Deposit Account No. 50-0585.

The attorney/agent of record invites the Examiner to contact him at (310) 557-2292 if the Examiner believes such contact would advance the prosecution of the case.

Dated: May 28, 2004

By: Rabindranath Dutta

Rabindranath Dutta
Registration No. 51,010

Please direct all correspondences to:

Rabindranath Dutta
Konrad Raynes Victor & Mann, LLP
315 South Beverly Drive, Ste. 210
Beverly Hills, CA 90212
Tel: 310-557-2292
Fax: 310-556-7984